



The AppLeS

Parameter Sweep Template:

User-Level Scheduling Middleware for the Grid

Henri Casanova Francine Berman Graziano Obertelli

University of California at San Diego

Rich Wolski

University of Tennessee, Knoxville

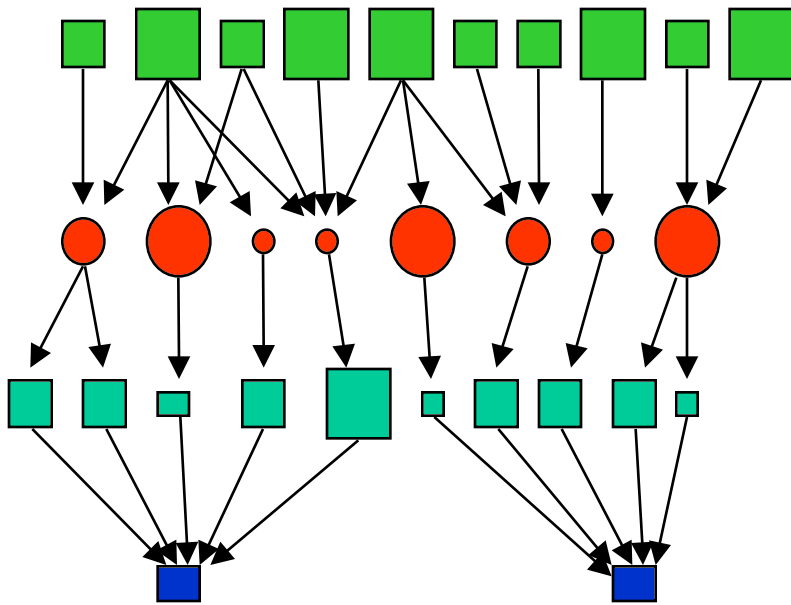
User-Level Middleware for the Grid

- Key for a usable Grid
- Targeted to a restricted application domain
- Directly targets end-user
- Hides the complexity of the Grid

An Application Domain: Parameter Sweeps

- Representative of large classes of applications:
 - Various simulations (Monte-Carlo, ...)
 - Parameter Searches
 - ...
- Used in various fields of science and engineering
- Large number of tasks, no task precedences in the general case \Rightarrow easy scheduling ?
 - I/O constraint
 - multiple stages of *post-processing*

Scheduling Issues



Scheduling Results

[1] *Heuristics for Scheduling Parameter Sweep Applications in Grid Environments*

H. Casanova, A. Legrand, D. Dzagorodnov, F. Berman (HCW'00)

Scheduling Algorithms
for PS Applications ?

Self-scheduling Algorithms

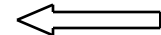
- workqueue
- workqueue w/ work stealing
- workqueue w/ work duplication
- ...

Easy to implement and quick
No need for performance predictions
Extremely adaptive
No planning (resource selection, I/O, ...)

Algorithms using Gantt charts: (using heuristics)

- MinMin, MaxMin
- Sufferage, XSufferage
- ...

More difficult to implement
Slower to run
Needs performance predictions
Tunable adaptivity
Heuristics for better planning



Simulation results in [1] show that:

- heuristics are worth it
- XSufferage seems like a good heuristic
- complex environments require better planning (Gantt chart)

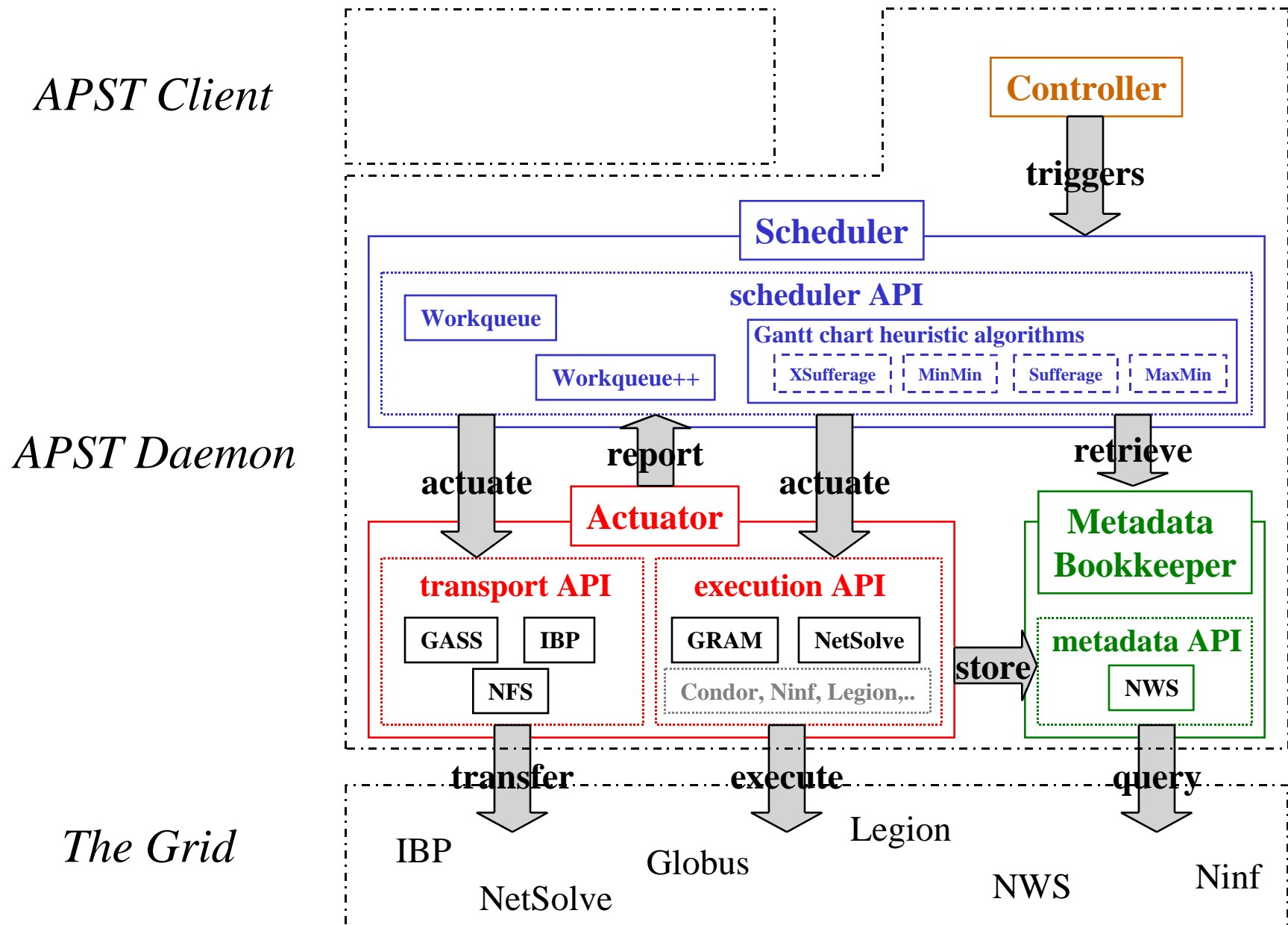
APST: User-Level Scheduling Middleware

- Design user-level middleware that:
 - is usable for real applications
 - is a vehicle for scheduling research
 - showcases the power of the Grid
- Challenges:
 - deployment
 - use multiple software infrastructures
 - isolate scheduler from deployment
 - scheduling
 - implement Gantt-chart heuristics
 - allow for multiple algorithms
 - adaptivity

Applications

- **INS2D:**
 - NASA Application
 - Fluid-dynamics
- **Mcell:**
 - Developed at the Salk Institute
 - Molecular modeling for Biology
- **Tphot:**
 - Developed at Univ. of Michigan / SDSC
 - Particle Physics
- **NeuralObjects:**
 - Developed at the NSI
 - Neuroscience
- **CS Simulation Applications for our own research:**
 - Grid simulation for application-level scheduling
 - Long-range CPU forecasting
 - Bricks
 - ...

APST Design



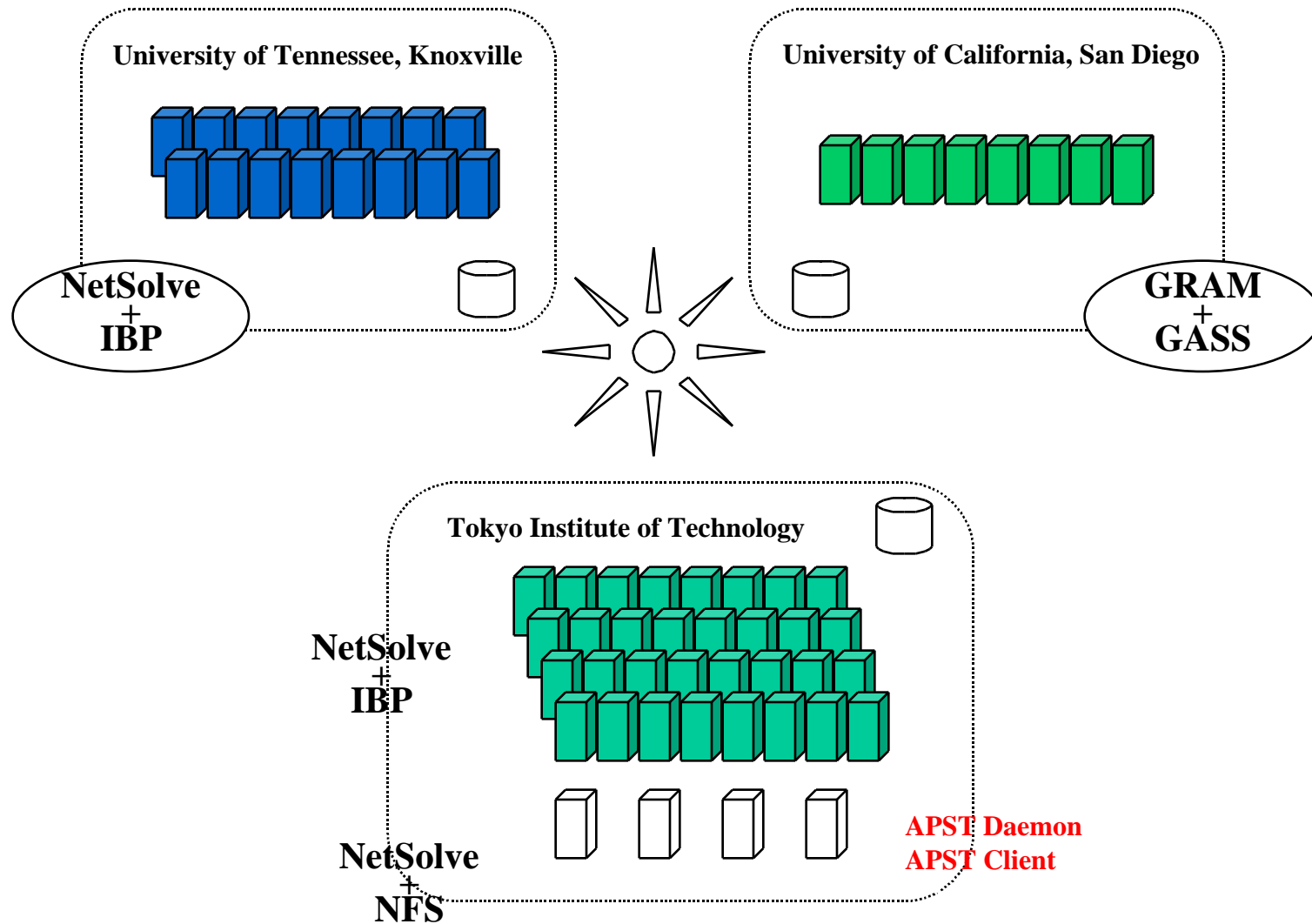
APST Implementation

- Actuator's APIs are **interchangeable** and **mixable**
 - (NetSolve+IBP) + (GRAM+GASS) + (GRAM+NFS)
- Scheduler API: can be replaced at run-time
 - allows for dynamic adaptation (workqueue++ Gantt chart)
 - still work in progress
- Use of multi-threading at multiple levels
 - latency hiding
- No Grid software is **required**
 - no NWS may lead to poor scheduling
 - no GASS/IBP may lead to poor performance
 - ...
- About 22,000 lines of C
 - 40% scheduling, 20% Grid interface, 40% Daemon/Client

APST and Globus

- Globus 1.1.3 (used via C APIs)
- At the moment: GRAM and GASS only
- Future versions will use:
 - MDS (now using a simple file)
 - HBM
- GASS wish-list
- Deployment on larger Globus testbeds

An Experimental Setting



Initial Experimental Results

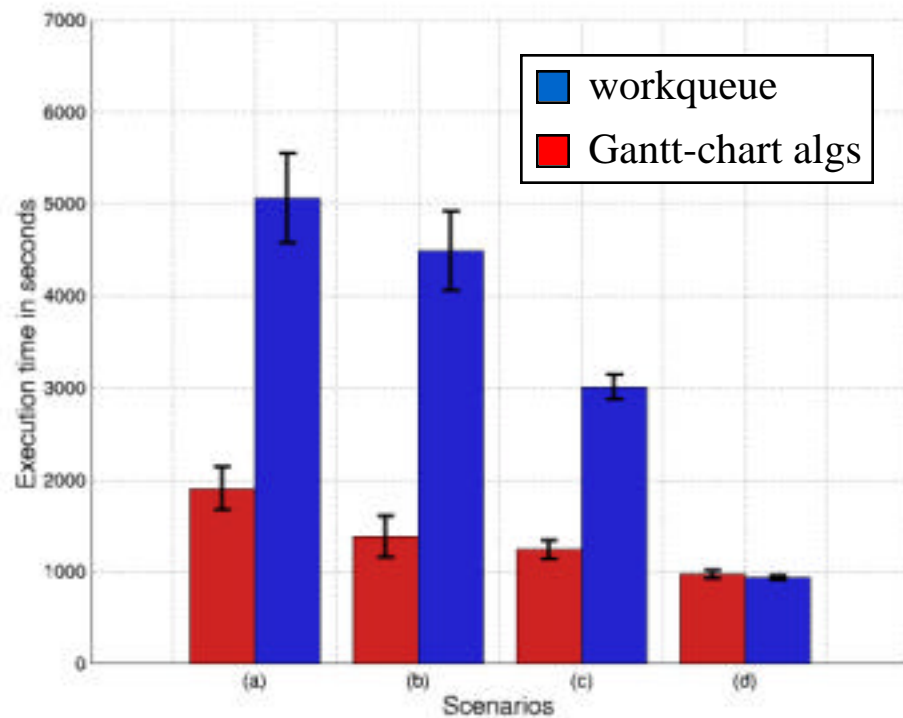
Experimental Setting:

Mcell simulation with 1,200 tasks:

- composed of 6 Monte-Carlo Simulations
- input files: 1, 1, 20, 20, 100, and 100 MB

4 scenarios: Initially

- (a) all input files are only in Japan
- (b) 100MB files replicated in California
- (c) in addition, one 100MB file replicated in Tennessee
- (d) all input files replicated everywhere



Related Work

- Nimrod at Monash Univ
- Condor at U. Wisc
- ILAB at NASA

References

[2] *The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid*

H. Casanova, G. Obertelli, F. Berman, R. Wolsi (SuperComputing'00)



<http://apples.ucsd.edu/apst>



Next Year's goals

- Release of APST v0.1:
 - To PACIs/NASA
 - Within the next 2 months
- Add Condor support
- Improve the Globus integration (MDS, HBM)
- Enhance the user interface's capabilities
- More flexible data management
- Generalize Grid model for scheduling algs.

Long-Range Goals

- Dynamic and automatic scheduling algorithm selection
- Enable efficient output post-processing
- Better performance prediction:
 - combining NWS and APST metadata
 - long-range forecasts
- New scheduling algorithms
- Computational Steering